

Opinnäytetyö (AMK)

Tietojenkäsittely

Tietoliikenne

2013

Panu Parkkila

GPS- PAIKANNUSOHJELMISTON SUUNNITTELU, TOTEUTUS JA TESTAUS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietojenkäsittely | Tietoliikenne

Toukokuu 2013 | 37 sivua

Esko Vainikka

Panu Parkkila

GPS-PAIKANNUSSOVELLUKSEN SUUNNITTELU, TOTEUTUS JA TESTAUS

Tämä opinnäytetyö käsittelee GPS-satelliittipaikannusjärjestelmää. Opinnäytetyön tavoitteena oli kehittää GPS:ää hyödyntävä paikannussovellus.

Opinnäytetyön teoriaosuudessa perehdytään aluksi lyhyesti GPS:n historiaan, hyödyntämistapoihin ja toimintaperiaatteisiin. Lisäksi teoriaosuus pohjustaa empiiristä osuutta perehdyttämällä käytettyihin työkaluihin.

Empiirinen osa sisältää GPS-paikannussovelluksen suunnittelun ja toteutuksen. Valmista sovellusta on testattu käytännössä sen toimintavarmuuden kartoittamiseksi. Opinnäytetyön viimeinen osa sisältää pohdintaa siitä, kuinka GPS-paikannussovellusta voisi jatkokehittää ja mitä parannuksia jatkokehitystoimenpiteet edellyttävät.

Opinnäytetyö soveltuu luettavaksi mobiilisovelluskehityksestä kiinnostuneille. Nykyaikaiset älypuhelimet tarjoavat suuren määrän erilaisia sensoreita ja komponentteja hyödynnettäväksi osana sovelluksia. App Inventor for Androidin kaltaisia työkaluja avuksi käyttäen on helppoa ja hauskaa perehtyä mobiilisovelluskehityksen alkeisiin.

ASIASANAT:

GPS, Android, App Inventor for Android, Paikannusohjelmisto

BACHELOR'S THESIS | ABSTRACT

TURKU UNIVERSITY OF APPLIED SCIENCES

Business Information Technology | Data communications

May 2013 | 37 pages

Esko Vainikka

Panu Parkkila

DESIGN, IMPLEMENTATION AND TESTING OF GPS LOCATION SOFTWARE

The topic of this thesis is GPS or Global Positioning System. The objective of the thesis is to develop a location software that uses GPS.

The theoretical part of this thesis begins with a brief history of GPS, the ways that it can be used and the working principles of GPS. Furthermore the theoretical part acts as a primer to the empirical part exploring the tools used in it.

The empirical part of this thesis contains design and implementation of a location software. The completed software has been tested in practice to survey the reliability of the software. The final part of the thesis discusses how the application could be developed further and what kind of improvements are required for further development.

The thesis is suitable to be read by anyone interested in the mobile application development. The modern smart phones offer a great amount of different kinds of sensors and components that can be used as a part of applications. By using tools like App Inventor for Android it is both easy and fun to become familiar with the basics of mobile application development.

KEYWORDS:

GPS, Android, App Inventor for Android, Location software

SISÄLTÖ

KÄYTETYT LYHENTEET JA SANASTO	6
1 JOHDANTO	7
2 GPS YLEISESTI	8
2.1 GPS:n historia	8
2.2 Sovelluksia ja käyttötarkoituksia	9
2.3 Muita satelliittipaikannusjärjestelmiä	11
3 GPS:N TOIMINTAPERIAATTEET	12
3.1 Segmentit	12
3.2 GPS:n toimintamalli	13
3.3 A-GPS	14
3.4 GPS-signaalien häiriöt	14
4 OHJELMISTON SUUNNITTELU	16
5 VALITUT LAITTEET JA MENETELMÄT	18
5.1 Android käyttöjärjestelmä	18
5.2 Samsung Galaxy S II Plus	18
5.3 App Inventor for Android	19
5.4 Google App Engine	20
5.5 TinyWebDB	20
5.6 Bing Maps	20
6 OHJELMISTON TOTEUTUS	21
6.1 Paikannussovellus	21
6.2 Tietokanta	25
6.3 Seurantasovellus	27
7 OHJELMISTON TESTAUS	29
7.1 Testaussuunnitelma	29
7.2 Kehitysversioiden testaaminen	29
7.3 Testaus kävellen	30
7.4 Testaus ajoneuvoissa	31

8 POHDINTA JA KEHITYSIDEAT	34
LÄHTEET	36

KUVAT

Kuva 1. Ruudunkaappaus Lissu Liikenteenseuranta -palvelusta. Numerot symboloivat bussilinjojen numeroita. (Tampereen joukkoliikenne 2013)	10
Kuva 2. App Inventorin käyttöliittymä.	19
Kuva 3. Mobiilisovelluksen käyttöliittymä App Inventorissa.	21
Kuva 4. Näkymättömät komponentit.	22
Kuva 5. Palikkaeditori.	22
Kuva 6. Painikkeiden toiminta palikkaeditorissa.	23
Kuva 7. Paikannussensorin toiminta palikkaeditorissa..	24
Kuva 8. Ajastin- ja ilmoituskomponenttien toiminta palikkaeditorissa.	25
Kuva 9. TinyWebDB-tietokannan vieminen Google App Engine-pilvipalveluun Google App Engine-launcherin avulla.	25
Kuva 10. Paikannussovelluksen lähettämiä koordinaatteja TinyWebDB-tietokannan taulukossa.	26
Kuva 11. Unix-skripti.	27
Kuva 12. Web-sovelluksen alkuosa.	27
Kuva 13. Web-sovelluksen PHP-osuus.	28
Kuva 14. Web-sovelluksen loppuosa.	28
Kuva 15. Sovelluksen kehitysversion testaaminen jalan.	29
Kuva 16. Ensimmäinen testaus jalan.	30
Kuva 17. Toinen testaus jalan.	30
Kuva 18. Sovelluksen testaus henkilöautossa.	32
Kuva 19. Sovelluksen testaus junassa.	33

KUVIOT

Kuvio 1. GPS-paikannusperiaate trilateraation avulla. (My-car-computer.com 2012)	13
--	----

KÄYTETYT LYHENTEET JA SANASTO

API	Application Program Interface. Ohjelmointirajapinta on määritelmä, jonka avulla eri ohjelmat voivat hyödyntää toisiaan (SearchExchange 2010).
Emulaattori	Sovellus, jonka avulla voidaan virtuaalisesti imitoida erilaisia ympäristöjä, kuten esimerkiksi älypuhelinta tai pelikonsolia (TechTerms.com 2008).
HTML	Hypertext Markup Language. Merkintäkieli, jota käytetään web-sivujen esittämiseen (SearchSOA 2005a).
JavaScript	Skripti- ja ohjelmointikieli, jota käytetään pääasiassa Internet-sovelluksissa (SearchSOA 2005b).
PHP	Hypertext Preprocessor. Internetskriptikieli, jota käytetään pääasiassa Internet-sovelluksissa. PHP-koodi suoritetaan palvelimella, joten se ei ole ulkopuolisten nähtävissä (SearchEnterpriseLinux 2006).
Pilvipalvelu	Termi, jota käytetään kuvaamaan mitä tahansa palvelualustaa, jota tarjotaan Internetin kautta. (SearchCloudProvider 2011).
WYSIWYG	What You See Is What You Get. Termi, jota käytetään kuvaamaan tietokoneohjelmia, joiden työjälki on nähtävissä koko ajan työn edetessä (Rouse, M. 2011).

1 JOHDANTO

GPS-paikannusta voidaan hyödyntää monilla eri tavoilla osana erilaisia liike-elämän toimintoja. Sen avulla voidaan automatisoida toimintoja, jotka muuten täytyisi suorittaa manuaalisesti. GPS-paikannuksen käyttö on arkipäiväistynyt 2000-luvulla huomattavasti, ja sitä hyödynnetään runsaasti niin tie- ja vesiliikenteessä kuin osana luonnossa liikkumista.

Opinnäytetyön tavoitteena on perehtyä GPS-satelliittipaikannusjärjestelmän toimintaan ja hyödynnettävyyteen sekä suunnitella, toteuttaa ja testata GPS-paikannussovellus. Teoriaosuudessa on kirjallisuutta ja Internet-lähteitä apuna käyttäen selvitetty, mitä GPS-satelliittipaikannusjärjestelmä sisältää, kuinka se toimii ja mitä käyttötarkoituksia sillä on. Lisäksi opinnäytetyö sisältää teoriaa opinnäytetyössä tehdyn paikannussovelluksen toteutuksessa käytetyistä menetelmistä. Empiirinen osa sisältää kuvauksen paikannussovelluksen suunnitelmasta, sovelluksen toteutuksesta vaihe kerrallaan sekä raportin sovelluksen testauksesta.

Suurimmat syyt opinnäytetyön aiheen valintaan olivat GPS-paikannusjärjestelmän tarjoamat lukemattomat hyödyntämismahdollisuudet ja henkilökohtainen kiinnostus mobiilisovelluskehitystä kohtaan.

2 GPS YLEISESTI

2.1 GPS:n historia

GPS-paikannusjärjestelmä (engl. Global Positioning System) on Yhdysvaltojen asevoimien alunperin sotilaskäyttöön suunnittelema globaali reaaliaikainen satelliittipaikannusjärjestelmä. GPS:n kehitys alkoi 1970-luvulla, kun Yhdysvaltojen meri- ja ilmavoimien 1960-luvulla alkaneet satelliittipaikannushankkeet yhdistettiin yhdeksi hankkeeksi, joka sai nimen NAVSTAR (Navigation System with Time and Ranging). Sotilaskäytössä GPS mahdollistaa esimerkiksi täsmäaseiden käytön sekä helpottaa taistelutilanteiden seuraamista. (Airos ym. 2007, 19.)

GPS:n vapautuminen siviilikäyttöön tapahtui asteittain. Sen tarpeellisuus siviilikäytössä alettiin ymmärtää varsin ikävällä tavalla, kun neuvostoliittolainen hävittäjä pudotti kurssiltaan harhautuneen Korean Air Linesin Boeing 747-matkustajalentokoneen Neuvostoliiton ilmatilassa vuonna 1983. Tapauksen jälkeen presidentti Reagan määräsi GPS:n vapautettavaksi myös siviilien käyttöön, kun järjestelmä on saavuttanut täyden toimintakunnon. (James. 2009.)

Vaikka GPS olikin siviilikäytössä, siviileille saatavilla olevaa signaalitaajuutta häirittiin pitkään siten, etteivät siviilikäyttöiset GPS-vastaanottimet kyenneet yhtä tarkkaan paikannukseen kuin sotilaskäytössä olevat vastaanottimet. GPS:n SA-häirintä poistettiin käytöstä vuonna 2000. Kentuckyssä Yhdysvalloissa sijaitsevan tarkkailuaseman tekemän testiraportin mukaan GPS:n siviilivastaanottimien keskimääräinen paikannustarkkuus parani SA-häirinnän loputtua välittömästi 45 metristä 6,3 metriin. (GPS.gov 2012.)

SA-häirintä (engl. Selective Availability) oli ominaisuus, jolla vuoteen 2000 asti häirittiin tahallisesti siviileille saatavilla olevaa GPS:n signaalitaajuutta.

Häirinnän tavoitteena oli rajoittaa siviililaitteiden paikannustarkkuutta. (Airos ym. 2007, 19.)

2.2 Sovelluksia ja käyttötarkoituksia

GPS tarjoaa lukemattomia käyttökohteita kaupallisissa ja vapaa-ajan tarkoituksissa. GPS-järjestelmän etuja on, että toiminta on kuluttajille veloituksetonta, ympärivuorokautista ja normaaleista sääolosuhteista riippumatonta.

SA-häirinnän loputtua ja paikannustarkkuuksien parannuttua erilaiset GPS-paikannusta hyödyntävät laitteet ovat yleistyneet ja GPS:n merkitys siviilikäytössä on kasvanut. Arvion mukaan vuonna 2010 myytiin noin 40 miljoonaa GPS-laitetta. Määrä ei pidä sisällään GPS-ominaisuudella varustettuja älypuhelimia, joiden on tutkimusyhtiö Berg Insightin mukaan arvioitu korvaavan tulevaisuudessa yhä enemmän perinteisiä GPS-laitteita. (O'Brien. 2010.)

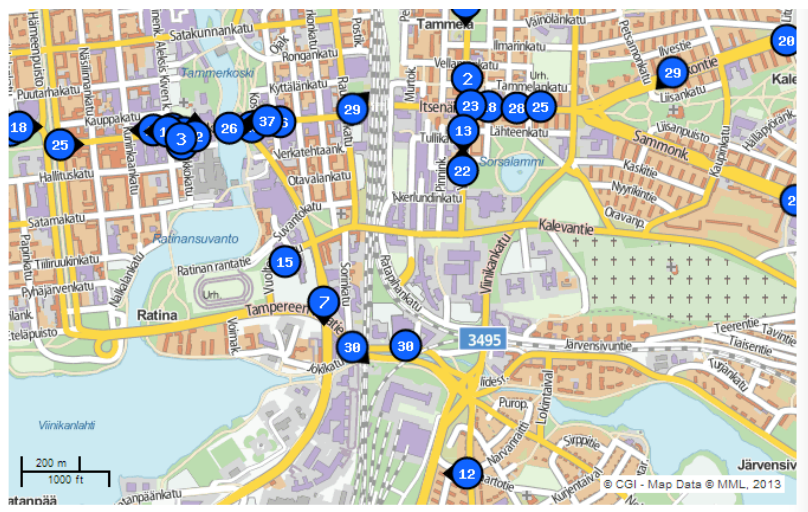
GPS:ää on mahdollista hyödyntää monilla eri tavoilla vapaa-ajan käyttötarkoituksiin. Sen avulla voi pitää kirjaa omista lenkkeily- ja pyöräilyreiteistään tai suunnistaa luonnossa. Lisäksi GPS-paikannuksen ympärille on syntynyt harrastus nimeltä geokätköily. Geokätköilyn harrastajat voivat etsiä toisten geokätköilijöiden jättämiä geokätköjä, joiden sijainteja on saatavilla Internetissä. Geokätkön koordinaatit syötetään GPS-paikantimeen, minkä jälkeen voi lähteä suunnistamaan kohti kätköä. (Luontoon.fi 2012.)

Logistiikkayrityksille suunniteltujen GPS-järjestelmään pohjautuvien ajoneuvoseurantajärjestelmien tavoitteina on muun muassa automatisoida ajopäiväkirjan pitäminen, mahdollistaa kuljetuskaluston etäseuranta ja seurata kuljettajan ajotapoja. Näillä seurantajärjestelmillä logistiikkayritykset saavat tehostettua toimintatapojaan automatisoimalla toimenpiteitä, jotka muuten olisi tehtävä manuaalisesti. (Navicom 2013.)

Ehkä yleisin siviilikäytössä oleva GPS:ää hyödyntävä sovellus on autonavigaattori. GPS:n käyttö tieliikenteessä ei rajoitu ainoastaan navigaattoreihin ja automaattisiin ajopäiväkirjoihin, vaan GPS mahdollistaa lisäksi taloudellisemman ajamisen myös henkilöautoilla. Hybridiautoihin suunniteltu sovellus pystyy GPS:ää apuna käyttäen ennustamaan tulevia tieosuuksia, kuten mäkiä ja ruuhkaisia kohtia, ja näin säätelemään älykkäämmin, milloin hybridauto käyttää sähkömoottoria bensiinimoottorin sijaan. (Sulopuisto. 2009.)

Suomessa VR tarjoaa mahdollisuuden seurata matkustajajunien reaaliaikaista sijaintia kartalla. Palvelu mahdollistaa sekä lähi- että kaukoliikenteen junien seuraamisen. Junat kartalla –palvelun toiminta perustuu junissa oleviin GPS-paikantimiin. Palvelun käyttö on mahdollista web-selaimen avulla, ja lisäksi tarjolla on iOS-, Android-, Symbian- ja Windows Phone -sovellukset. (Junat kartalla – VR 2011.)

Julkisen liikenteen reaaliaikaisia paikkatietoja hyödynnetään Suomessa ainakin Tampereella ja Helsingissä. Lissu Liikenteenseuranta –palvelu mahdollistaa Tampereen joukkoliikenteen linja-autojen sijainnin seuraamisen kartalla. Lisäksi paikkatietoja hyödynnetään bussipysäkeiden näytöiltä, joilla näkyy arvioitu linja-auton saapumiseen kuluva aika. (Tampereen joukkoliikenne 2013.)



Kuva 1. Ruudunkaappaus Lissu Liikenteenseuranta -palvelusta. Numerot symboloivat bussilinjojen numeroita (Tampereen joukkoliikenne 2013).

2.3 Muita satelliittipaikannusjärjestelmiä

Venäjän kehittämä satelliittipaikannusjärjestelmä on nimeltään GLONASS. Sen ensimmäinen satelliitti laukaistiin maan kiertoradalle vuonna 1982, mutta se saavutti täyden kattavuuden hitaasti, vasta vuosina 1994-1995 laukaistujen satelliittien jälkeen. GLONASS-järjestelmän heikko toimintavarmuus verrattuna GPS:ään johtuu osittain siitä, että GLONASS-satelliittien käyttöikä on lyhyt, sekä siitä, että GLONASS-järjestelmän ylläpitoa laiminlyötiin pitkään 1990-luvulla. GLONASS-satelliittien kiertoradat on optimoitu siten, että järjestelmä saavuttaa parhaan kattavuuden Venäjän alueella, mutta muissa maapallon osissa se toimii huonommin kuin GPS. (Airos ym. 2007, 26-27.)

Galileo on Euroopan Avaruusjärjestön ja Euroopan Unionin yhteishankkeena kehittämä satelliittipaikannusjärjestelmä, joka on ensisijaisesti suunniteltu siviilikäyttöön. Eurooppalaisten tarve kehittää oma ja muista riippumaton satelliittipaikannusjärjestelmä syntyi, koska GPS ja GLONASS on molemmat kehitetty ensisijaisesti sotilaskäyttöön, ja koska niiden hallinta on täysin Yhdysvaltojen ja Venäjän käsissä. Galileon kehitys alkoi 1994 ja sen ensimmäinen testisatelliitti laukaistiin maan kiertoradalle joulukuussa 2005. (Airos ym. 2007, 32-34.)

Kiinan kansantasavallan satelliittipaikannusjärjestelmä on nimeltään Compass, ja se pohjautuu kiinalaisten aiemmin rakentamaan Beidou-satelliittipaikannusjärjestelmään. Compassin satelliittiverkoston on määrä saavuttaa maailmanlaajuinen kattavuus vuoteen 2020 mennessä, ja sen on määrä käsittää vähintään 30 satelliittia. Compass-järjestelmän satelliiteilla on ominaisuus, jota ei muiden satelliittipaikannusjärjestelmien satelliiteista löydy: ne pystyvät lähettämään tekstiviestejä. Tämä ominaisuus mahdollistaa tekstiviestien lähettämisen Compass-päätelaitteen avulla alueelta, jolla ei ole matkapuhelinverkkoa. (GPS Daily 2010.)

3 GPS:N TOIMINTAPERIAATTEET

3.1 Segmentit

GPS-satelliittipaikannusjärjestelmä koostuu kolmesta segmentistä eli osasta. Segmentit ovat avaruus-, hallinta- ja käyttäjäsegmentti. (Airos, Korhonen, Pulkkinen, 2007, 20.)

GPS-paikannusjärjestelmän avaruussegmentti käsittää tällä hetkellä kaikkiaan 32 satelliittia (U.S. Naval Observatory 2013). Jotta GPS-paikannusjärjestelmä toimisi jatkuvasti joka puolella maapalloa, se vaatii vähintään 24 toimintakuntoista GPS-satelliittia. GPS-satelliittien lentoradat on järjestetty siten, että niiden kiertoaika on 12 tuntia, jolloin sama satelliitti ehtii kiertämään maapallon kaksi kertaa vuorokaudessa. GPS-satelliittien lentoradoissa on otettu huomioon myös se, että GPS-paikantimella on jatkuvasti mahdollisuus vastaanottaa signaaleja vähintään neljästä satelliitista. (Airos, Korhonen, Pulkkinen, 2007, 20.)

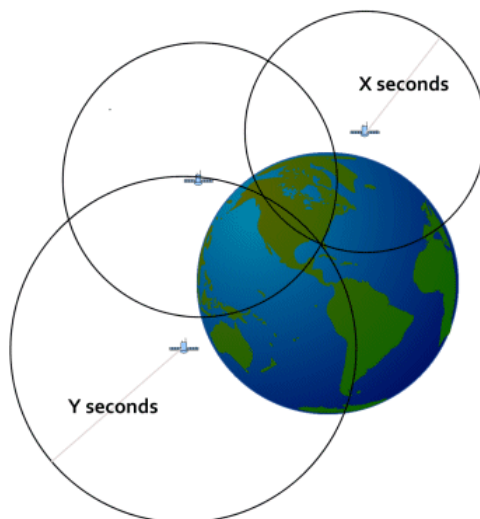
GPS:n toimivuuden takaamiseksi GPS-järjestelmää valvotaan ja ylläpidetään maasta. Järjestelmän valvonnan vastuu on Yhdysvaltojen ilmavoimilla, jotka valvovat sen toimintaa päätarkkailuasemalta, joka sijaitsee Coloradossa, Yhdysvalloissa. Päätarkkailuasemalla on lisäksi tukenaan ympäri maailmaa sijoitettuja passiivisia tarkkailuasemia, jotka välittävät vastaanottamansa tiedot päätarkkailuasemalle. Asemalla tarkkaillaan GPS-satelliittien lähettämiä signaaleja, jotka sisältävät muun muassa satelliitin paikan kiertoradallaan sekä sen kuntotilanteen. (Ogaja, 2011, 7-8.)

Erilaiset GPS-vastaanottimet muodostavat GPS-järjestelmän käyttäjäsegmentin. GPS:n suosio siviilikäytössä on kasvanut runsaasti SA-häirinnän loputtua. On arvioitu, että 90 % kaikista GPS-vastaanottimista on siviilikäytössä. (Ogaja 2011, 4.)

3.2 GPS:n toimintamalli

Jokainen GPS-satelliitti lähettää signaalia kahdella taajuudella, joiden lyhenteet ovat L1 ja L2. L1-taajuus on siviilikäytössä, mutta L2-taajuuden signaali on varattu sotilaskäyttöön. Ionosfäärissä tapahtuvat ilmiöt, kuten revontulet, aiheuttavat signaalin nopeasta kulkuvauhdista johtuen hetkittäin suuriakin virheitä mittaustuloksissa. Kahden eri taajuudella saapuvan signaalin vastaanottaminen mahdollistaa tarkemman sijainnin määrittämisen, kun maapallon ilmakehän ionosfäärin aiheuttama viive saadaan eliminoitua mittaustuloksesta. (Airos ym. 2007, 15, 21.)

GPS-vastaanotin laskee sijaintinsa trilateraation eli kolmipistemittauksen avulla. Vastaanotettuaan signaalit GPS-satelliiteilta GPS-vastaanotin määrittää niiden sisältämän informaation perusteella oman sijaintinsa laskemalla matka-ajan jokaiselta satelliitilta. Vastaanotin kykenee määrittämään vaakasuoran sijaintinsa kolmen satelliitin signaalien avulla, mutta määrittääkseen myös korkeutensa se vaatii signaalin myös neljänneltä satelliitilta. Mitä enemmän satelliitteja on saatavilla, sitä parempi vastaanottimen määrittämistarkkuus on. (Ogaja, 2011, 10.)



Kuvio 1. GPS-paikannusperiaate trilateraation avulla. (My-car-computer.com 2012)

3.3 A-GPS

A-GPS (engl. Assisted GPS) on älypuhelimissa oleva ominaisuus, joka parantaa perinteisen GPS:n toimintanopeutta. A-GPS-ominaisuudella varustettu älypuhelin hakee aluksi summittaisen sijaintitietonsa esimerkiksi matkapuhelinverkon tukiasemalta, minkä jälkeen se kykenee nopeammin tulkitsemaan GPS-satelliiteilta vastaanottamiaan signaaleja ja laskemaan sijaintinsa. Lisäksi A-GPS-ominaisuuden ansiosta matkapuhelin kykenee tarkempaan paikannustarkkuuteen olosuhteissa, joissa ympäristön rakenteet häiritsevät GPS-signaalien vastaanotettavuutta. (Windows Phone Central 2009.)

3.4 GPS-signaalien häiriöt

GPS-satelliittien signaalit ovat erittäin häiriöalttiita. Tämä johtuu siitä, että GPS-satelliittien keskimääräinen välimatka vastaanottiin on noin 20 000 kilometriä ja siitä, että satelliitin lähettämän GPS-signaalin vahvuus on teholtaan hyvin heikko. Näin ollen GPS-vastaanottimen saamaa signaalia voidaan häiritä tahallisesti tai tahattomasti pienitehoisillakin häirintälaitteilla. (Airos ym. 2007, 40.)

GPS-signaalien vastaanotossa syntyy häiriöitä, kun samalla alueella olevan radiolähettimen taajuusalue on lähellä GPS-satelliittien signaalien taajuuksia. Muun muassa radioamatöörit ja satelliittipuhelimet käyttävät radiotaajuuksia, jotka ovat lähellä GPS-satelliittien lähettämiä taajuuksia. (Airos ym. 2007, 44.)

Eräs esimerkkitapaus tarkoituksella aiheutetusta GPS-häirinnästä tapahtui vuonna 2009, kun Newarkin lentokentällä Yhdysvalloissa havaittiin toistuvia häiriöitä lentokoneiden ja lennonjohdon satelliittipaikannusta hyödyntävissä järjestelmissä. Häiriöiden syyksi paljastui lopulta kuorma-autonkuljettaja, jolla oli autossaan edullinen GPS-häirintälaitte. Tällaisella edullisella GPS-häirintälaitteella voidaan estää esimerkiksi satelliittipaikannukseen pohjautuvan ajoneuvoseurantajärjestelmän toiminta, mutta halvat ja helposti saatavilla olevat

GPS-häirintälaitteet aiheuttavat häiriöitä myös muille alueella toimiville GPS-vastaanottimille. Erityisen vaarallisia ovat GPS-häirintälaitteet, jotka lähettävät virheellisiä signaaleja sen sijaan, että vain häiritsisivät signaalin luettavuutta (Reuters 2012).

4 OHJELMISTON SUUNNITTELU

Opinnäytetyön tavoitteena oli kehittää GPS-paikannusjärjestelmää hyödyntävä paikannusohjelmisto. Ohjelmistosta oli tavoitteena tehdä mahdollisimman käyttökokemukseltaan mahdollisimman helppokäyttöinen. Määrittelin vaatimuksiksi paikannuslaitteen sovellukselle kaksi ominaisuutta; ohjelma käyttää laitteen paikannussensoria paikantaakseen itsensä ja sen jälkeen lähettää oman olinpaikkansa leveys- ja pituusasteiden koordinaatit web-palvelimelle.

Pohdin pitkään, toteutanko järjestelmän tarkoitusta varten rakennetun GPS-vastaanottimen vai A-GPS tuella varustetun älypuhelimien avulla. Valintaan vaikuttavat eniten laitteen edullinen hinta sekä tekniikan hyvä toimintavarmuus. Asiaan perehdyttyäni päätin toteuttaa paikantimen älypuhelimien avulla.

Itse tehdyn GPS-laitteen rakentaminen olisi vaatinut useita komponentteja: GPS-vastaanottimen, Arduino-mikrokontrollerin, virtalähteen sekä suojakuoren (Ladyada.net 2011). Kyseinen kokoonpano olisi riittänyt ainoastaan GPS-lokin pitoon ja tallentanut GPS-koordinaatit muistikortille, josta ne olisi voinut myöhemmin syöttää tietokoneelle. Jotta olisin päässyt tarkkailemaan GPS-paikantimen liikkeitä reaaliaikaisesti, olisi pitänyt lisäksi hankkia GPRS-moduuli, joka lähettäisi paikkatiedot Internet-yhteyden avulla web-palvelimelle. GPRS-moduuli vaaditaan Internet-yhteyden saamiseksi.

GPS-vastaanottimen komponenttien hinnat Cooking Hacks verkkokaupasta ovat:

- Arduino Uno Rev.3 -mikrokontrolleri 20€ (Cooking Hacks 2013a).
- 3G/GPRS-moduuli 149€ (Cooking Hacks 2013b).
- 7.4V LiPo 2200mAh -akku 20€ (Cooking Hacks 2013c).
- Arduino Project Enclosure -suojakotelo 9.90€ (Cooking Hacks 2013d).

Kokonaishinnaksi näille komponenteille, sisältämättä pakollisia lisätarvikkeita, kuten transistoreja, kondensaattoreita sekä signaalin vastaanottimen vahvistinta, olisi tullut toimituskuluineen noin 200€. Vertailuksi halvin A-GPS – tuella varustettu Android-älypuhelin (LG Optimus L5) maksoi Verkkokauppa.comissa keväällä 2013 alle 170 euroa, tarjoten kaikki samat vaaditut ominaisuudet kuin itse rakennettu GPS-vastaanotin (Verkkokauppa.com 2013). Lisäksi GPS-paikantimen toiminnan tilan seuranta olisi ollut erityisen hankalaa ilman erillistä näyttökomponenttia.

Halusin toteuttaa GPS-vastaanottimen sijainnin seuraamiseen tarkoitetun sovelluksen siten, että sitä pystyisi käyttämään mahdollisimman monella eri laitteella. Selaimen kautta käytettävä seurantaohjelma vaikutti kaikkein luontevimmalta valinnalta, koska näin GPS-vastaanottimen liikkeitä pystyy seuraamaan minkä tahansa tietokoneen, tabletin tai älypuhelimien näytöltä verkkoselaimen avulla. Selainpohjaisen seurantaohjelmiston etuja olivat paitsi laiteriippumattomuus, myös se, ettei se vaadi toimiakseen erillistä asennusta.

5 VALITUT LAITTEET JA MENETELMÄT

5.1 Android käyttöjärjestelmä

Valitsin GPS-vastaanottimen toteutusalueksi Googlen Android-käyttöjärjestelmää käyttävän matkapuhelimen. Päädyin käyttämään Android-puhelinta, koska Android on noussut suosituimmaksi älypuhelinkäyttöjärjestelmäksi, ja sen on uskottu säilyttävän johtavan asemansa ainakin Yhdysvalloissa vuoteen 2016 asti. Androidin markkinaosuus oli vuoden 2012 lopulla 68,3% (Eadicicco, L. 2012). Android-älypuhelimia valmistavat useat laitevalmistajat, joista tunnetuimpia ovat HTC, Samsung, SonyEricson, Huawei, Motorola ja ZTE (Android.fi 2010).

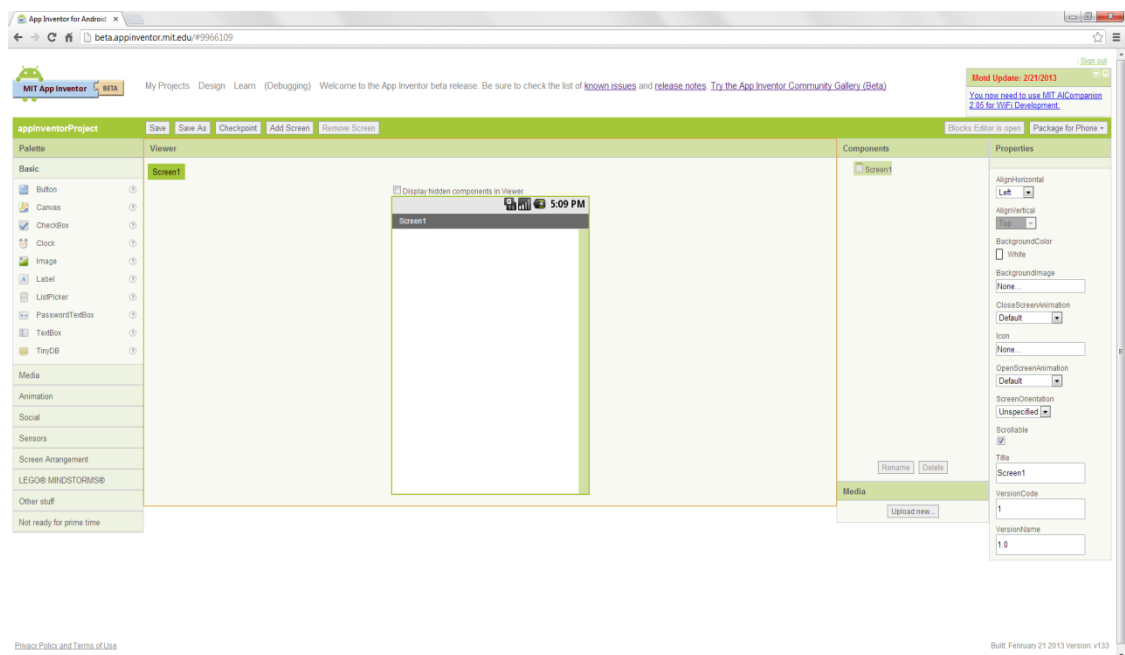
Android on Googlen kehittämä, Linuxiin pohjautuva käyttöjärjestelmä, joka julkistettiin marraskuussa 2007. Käyttöjärjestelmän uusin versio 4.2. Jelly Bean julkaistiin marraskuussa 2012, mutta keväällä 2013 suosituin Android-käyttöjärjestelmä oli helmikuussa 2011 julkaistu Gingerbread, joka oli käytössä yhä 44 %:ssa Android-laitteista. Toiseksi suosituin versio oli joulukuussa 2011 julkaistu Ice Cream Sandwich (28,6 %) ja vasta kolmantena Jelly Bean versiot 4.1 ja 4.2 (yhteensä 16,5%) (Android (operating system) 2013).

5.2 Samsung Galaxy S II Plus

Opinnäytetyön matkapuhelinohjelmiston testausalueena oli Samsungin valmistama Galaxy S II Plus –älypuhelin, johon oli asennettuna käyttöjärjestelmäksi Android 4.1.2 Jelly Bean. Galaxy S II Plus on Samsungin alkuvuodesta 2013 julkaisema päivitetty versio Samsungin aiemmasta Android-käyttöjärjestelmää käyttävästä älypuhelimesta Samsung Galaxy S II:sta. Galaxy S II Plus tarjoaa tuen A-GPS –paikannukselle, joten se oli sopiva vaihtoehto GPS-vastaanottimeksi (GSM Arena 2013).

5.3 App Inventor for Android

App Inventor for Android on alun perin Googlen kehittämä, mutta nykyään Massachusettsin teknillisen korkeakoulun ylläpitämä Java-pohjainen Android-ohjelmistokehitystyökalu. Se toimii täysin verkkoselaimen kautta, mutta edellyttää että Java on asennettu tietokoneelle. App Inventorilla luotuja sovelluksia on mahdollista testata Android-emulaattorin kanssa suoraan tietokoneella tai ladata ohjelma suoraan matkapuhelimeen.



Kuva 2. App Inventorin käyttöliittymä.

App Inventor on WYSIWYG-ohjelma, eikä sen käyttö vaadi ohjelmointikielen osaamista, mutta ohjelmointilogiikan ymmärtämisestä on paljon apua. Tämä mahdollistaa Androidin perusominaisuuksia hyödyntävien applikaatioiden kehittämisen, vaikka käyttäjä ei osaisikaan ohjelmoida. Toimiakseen App Inventor vaatii Google-tilin, jota ilman palveluun ei voi kirjautua.

App Inventorin käyttöliittymä koostuu kahdesta osa-alueesta: pääikkunasta ja palikkaeditorista (Blocks Editor). Pääikkunassa suunnitellaan ohjelmalle ulkoasu ja valitaan käytettävät komponentit. Palikkaeditorin avulla määritetään ohjelman

toiminnallinen osuus, eli mitä pääikkunassa luotujen komponenttien halutaan tekevän.

5.4 Google App Engine

Google App Engine on Googlen vuodesta 2008 lähtien tarjoama pilvipalvelu, jonka avulla on mahdollista käyttää Googlen tarjoamia ohjelmointirajapintoja. App Engine tarjoaa tuen useille kielille kuten Python, Java ja PHP. App Enginessä olevat sovellukset skaalautuvat sitä mukaa, kuinka suuria ne ovat ja kuinka paljon niitä käytetään (Google App Engine 2013).

5.5 TinyWebDB

TinyWebDB on tietokanta, jota voidaan käyttää App Inventorilla tehtyjen sovellusten datan tallentamiseen ja lukemiseen. Google tarjoaa järjestelmien testausta varten rakennetun TinyWebDB –tietokannan, joka on samanaikaisesti kaikkien App Inventorilla TinyWebDB:tä käyttävien käytössä. Testitietokannan merkintöjen määrä on rajoitettu tuhanteen merkintään, minkä jälkeen vanhin merkintä poistetaan automaattisesti. Omiin tarkoituksiin kapasiteetiltaan suuremman TinyWebDB:n rakentaminen on mahdollista Googlen App Engine –pilvipalvelua apuna käyttäen (MIT App Inventor 2013a).

5.6 Bing Maps

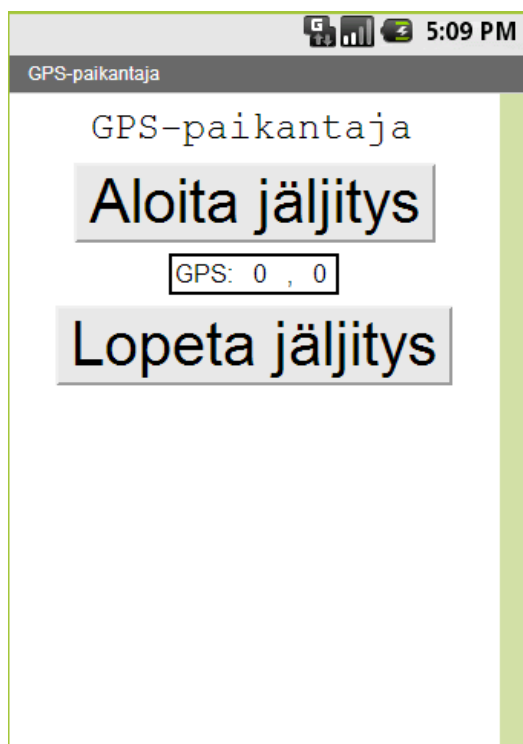
Bing Maps on Microsoftin karttapalvelu, jonka ohjelmointirajapinnan avulla on mahdollista luoda personoitu Bing Maps-kartta omalle web-sivulle tai sovellukselle. Kartan luominen omiin tarkoituksiin edellyttää sovellusavaimen hankkimista Bing Maps Account Centerin kautta. (Bing Maps Account Center 2013).

6 OHJELMISTON TOTEUTUS

6.1 Paikannussovellus

Aloitin ohjelmiston toteuttamisen tekemällä ensimmäiseksi paikannussovelluksen. Tutustuttuani ensiksi App Inventor for Androidin toimintoihin suunnittelin ensimmäiseksi käytettävän ohjelmiston käyttöliittymän ulkoasun. Tein siitä mahdollisimman yksinkertaisen, koska sen tarkoituksena oli ainoastaan paikkatiedon välittäminen web-palvelimen kautta seurattavaksi.

Paikannussovelluksen käyttöliittymä on pelkistetty, se koostuu kahdesta painikkeesta, joiden avulla jäljitystä säädellään, ja tekstikentästä, joka näyttää koordinaatit. Sovelluksen koordinaattitekstikenttä päivittyy sitä mukaa, kun puhelimen GPS-vastaanotin välittää sovellukselle uudet tiedot nykyisestä sijainnista. Kuva 3 on ruutukaappaus App Inventorin pääikkunasta ja siinä on nähtävissä sovelluksen valmis käyttöliittymä.



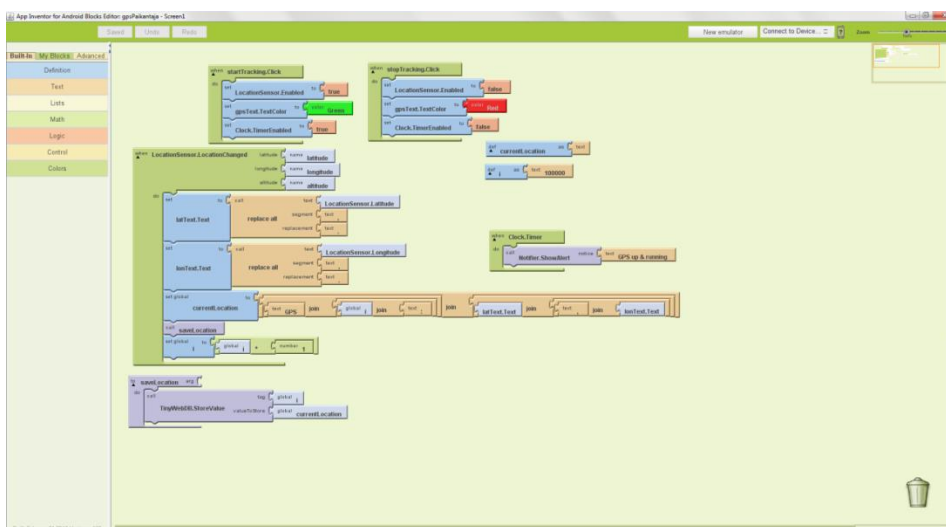
Kuva 3. Mobiilisovelluksen käyttöliittymä App Inventorissa.

Kuvassa 4 ovat sovelluksen näkymättömät komponentit, joiden toimintaa hallitaan palikkaeditorilla. Säädin paikannussensorikomponentin (LocationSensor) ominaisuudeksi, että se yrittää päivittää uuden sijainnin joka kymmenes sekunti. Mikäli paikannussensori ei saa uutta sijaintia, se ei muuta nykyistä sijaintia. TinyWebDB-komponentin palvelusoitteeksi (ServiceURL) määrittelin oman TinyWebDB-tietokannan, joka oli isännöity Googlen appspot.com -domainiin Googlen App Enginen kautta. Ilmoitus- ja ajastinkomponenttien tarkoituksena on antaa käyttäjälle ilmoitus siitä, että sovelluksen jäljitysominaisuus on kytkettynä puhelimen ollessa lukitustilassa. Ilmoitukset on määritetty App Inventorin pääikkunassa tulemaan aina viidentoista sekunnin välein.



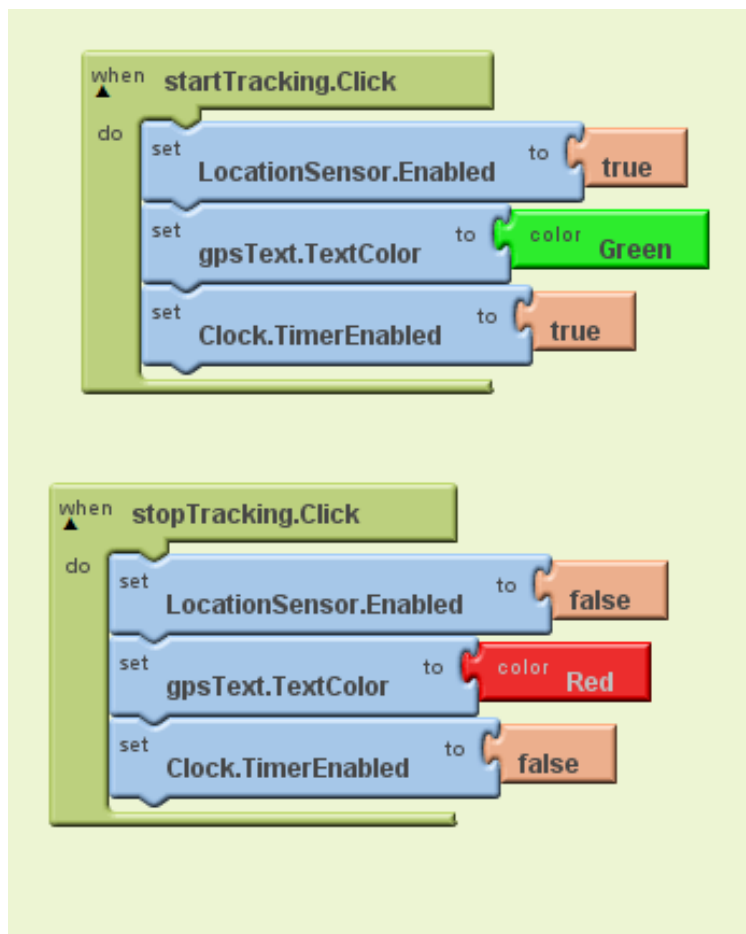
Kuva 4. Näkymättömät komponentit.

Kun App Inventorin pääikkunassa on lisätty tarvittavat komponentit ja suunniteltu sovelluksen ulkoasu, luodaan palikkaeditorilla sovelluksen toiminnallinen osuus. Kuvassa 5 on paikannussovelluksen palikkaeditorinäkymä, josta näkyy sovelluksen lopullinen toiminnallinen osuus.



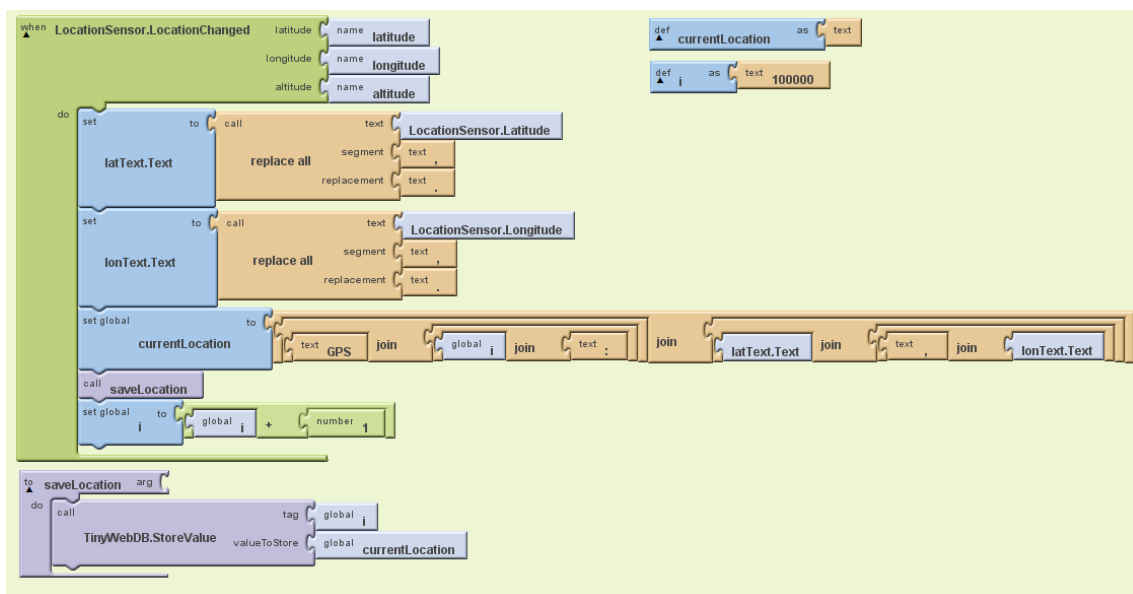
Kuva 5. Palikkaeditori.

Pääikkunassa määriteltyjen painikkeiden toiminta on järjestetty kuvan 6 esittämällä tavalla. Kuvasta käy ilmi, mitä tapahtuu aloitus- ja lopetuspainiketta painamalla. Kun painaa aloituspainiketta, jonka toiminta on kuvassa vasemmanpuoleisessa palikassa, antaa sovellus paikannussensorille ja kellokomponentille luvan käynnistyä. Aloituspainike muuttaa myös pääikkunassa luodun viimeisintä sijaintia esittävän tekstikentän GPS-tekstin vihreäksi. Lopetuspainiketta painamalla sovellus keskeyttää paikannussensorin ja kellokomponentin toiminnan, sekä muuttaa GPS-tekstin värin punaiseksi.



Kuva 6. Painikkeiden toiminta palikkaeditorissa.

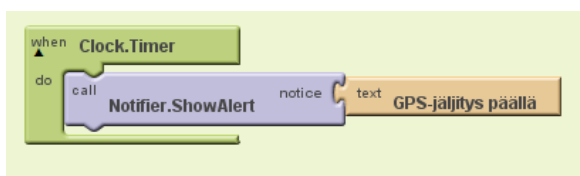
Älypuhelimien sisäänrakennetun GPS:ää ja mobiiliverkkopaikannusta hyödyntävän paikannussensorin toiminta sovelluksessa on havainnollistettu kuvassa 7. Sovellus alustaa muuttujan `currentLocation`, johon tallennetaan aina nykyisen sijainnin koordinaatit ja askelmuuttujan `i`, jonka avulla koordinaatit saadaan sijoitettua halutunlaiseen järjestykseen TinyWebDB-tietokantaan.



Kuva 7. Paikannussensorin toiminta palikkaeditorissa..

Paikannussensorin toiminta on järjestetty siten, että se toteuttaa aina samat toimenpiteet, kun paikannussensori saa uuden paikkatiedon. Kun uusi paikkatieto on saatu, sovellus muuttaa ensin koordinaattien pilkut pisteiksi, koska seurantasovelluksen kartassa pilkkua voi käyttää ainoastaan pituus- ja leveysasteiden erottimena. Tämän jälkeen sovellus näyttää nykyisen sijainnin pääikkunassa määritetyllä tekstikentällä sekä lähettää yhdistetyt pituus- ja leveysasteet, askelmuuttujalla `i` merkittynä, TinyWebDB-tietokantaan `saveLocation`-funktiota käyttäen. Lopuksi askelmuuttuja `i`:n arvoa kasvatetaan yhdellä, minkä jälkeen sovellus jää odottamaan paikannussensorilta jälleen uutta sijaintia ja suorittaa uudelle sijaintitiedolle samat toimenpiteet.

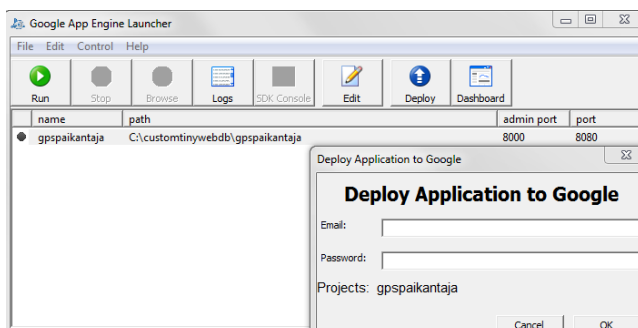
Kuvasta 8 käy ilmi ajastin- ja ilmoituskomponenttien toiminta. App Inventorin pääikkunassa ajastinkomponentin aikaväliksi on määritetty 15 sekuntia, eli kun paikannussovelluksen käyttäjä on painanut Aloita jäljitys –painiketta, antaa ajastinkomponentti määritetyin väliajoin ilmoituskomponentille luvan näyttää älypuhelimien näytöllä ilmoituksen. Ilmoituksen tekstiksi on palikkaeditorissa määritetty ”GPS-jäljitys päällä”. Tämän ilmoituksen tavoitteena on ilmoittaa käyttäjälle, että sovelluksen GPS-jäljitystoiminto on käynnissä, ja että älypuhelimien sijaintitietoja välitetään web-selaimessa nähtäväksi.



Kuva 8. Ajastin- ja ilmoituskomponenttien toiminta palikkaeditorissa.

6.2 Tietokanta


Määrittelin paikannussovelluksen lähettämään saamansa koordinaatit Googlen App Enginen kautta luotuun TinyWebDB-tietokantaan. Syötteiden saaminen kronologiseen järjestykseen onnistui määrittelemällä palikkaeditorissa tietokantaan syötettävien koordinaattien tunnisteiden alkamaan sadastatuhannesta. Alunperin yritin saada tunnisteiden alkamisluvuksi nollaa, mutta se johti siihen, että osa syötteistä meni väärään järjestykseen. Kuvassa 9 TinyWebDB:n lähdekoodi vietään Google App Engine –pilvipalveluun.



Kuva 9. TinyWebDB-tietokannan vieminen Google App Engine-pilvipalveluun Google App Engine-launcherin avulla.

Oman TinyWebDB-tietokannan perustaminen Googlen App Engine-pilvipalveluun edellytti palveluun kirjautumista omalla Google-tilillä, minkä jälkeen palveluun luotiin Dashboard-hallintapaneelista uusi projekti, jolle annoin nimeksi "gpspaikantaja". Kun uusi projekti oli luotu, käytin tietokannan pystytyksessä apuna Google App Engine Launcher -ohjelmaa, jolla sain lähetettyä App Inventorin web-sivuilta ladatun TinyWebDB-tietokannan lähdekoodin Googlen App Engine -pilvipalveluun.

Lähdekoodin saaminen oikeaan paikkaan pilvipalvelussa edellytti, että lähdekoodissa muokattiin projektin nimi vastaamaan Dashboard-hallintapaneelissa luotua nimeä. Kun lähdekoodi oli saatu vietyä pilvipalveluun, oli TinyWebDB-tietokanta valmis käytettäväksi. Kuvassa 10 on paikannusohjelman syötteillä populoitu TinyWebDB -taulukko.



App Inventor.

The page you are looking at is a web page interface for existing entries, and also delete individual entries

Available calls:

- [/storevalue](#): Stores a value, given a tag and a value
- [/getvalue](#): Retrieves the value stored under a given tag. Returns the empty string if no value is found.

Key	Value	Created (GMT)	
100218	"GPS100218: 61.50461 , 23.71853"	Tue Apr 23 13:11:22 2013	Delete
100217	"GPS100217: 61.50457 , 23.71842"	Tue Apr 23 13:11:11 2013	Delete
100216	"GPS100216: 61.50463 , 23.71828"	Tue Apr 23 13:11:01 2013	Delete
100215	"GPS100215: 61.50483 , 23.71825"	Tue Apr 23 13:10:51 2013	Delete
100214	"GPS100214: 61.50496 , 23.7184"	Tue Apr 23 13:10:41 2013	Delete
100213	"GPS100213: 61.50519 , 23.71845"	Tue Apr 23 13:10:01 2013	Delete
100212	"GPS100212: 61.50522 , 23.71828"	Tue Apr 23 13:09:51 2013	Delete
100211	"GPS100211: 61.50521 , 23.71802"	Tue Apr 23 13:09:41 2013	Delete
100210	"GPS100210: 61.50511 , 23.71734"	Tue Apr 23 13:09:29 2013	Delete
100209	"GPS100209: 61.50461 , 23.72004"	Tue Apr 23 13:09:16 2013	Delete
100208	"GPS100208: 61.50399 , 23.72256"	Tue Apr 23 13:09:06 2013	Delete
100207	"GPS100207: 61.50327 , 23.72438"	Tue Apr 23 13:08:56 2013	Delete
100206	"GPS100206: 61.50267 , 23.7254"	Tue Apr 23 13:08:46 2013	Delete
100205	"GPS100205: 61.50137 , 23.72704"	Tue Apr 23 13:08:36 2013	Delete
			Delete

Kuva 10. Paikannussovelluksen lähettämiä koordinaatteja TinyWebDB-tietokannan taulukossa.

6.3 Seurantasovellus

Seurantasovelluksen toteutin Microsoftin Bing Mapsiin pohjautuvana web-sovelluksena. Yksi seurantasovelluksen toteuttamisen suurimmista haasteista oli tietojen siirtäminen TinyWebDB-tietokannasta web-sivulla olevaan karttaan. Toteutin tietojen siirtämisen tekemällä seurantasovellusta isännöivälle Unix-palvelimelle skriptin, joka hakee päällä ollessaan 10 sekunnin välein tiedot TinyWebDB-tietokannasta, siivoaa tekstin haluttuun muotoon sekä tallentaa tiedot edelleen Unix-palvelimelle tiedostoon, josta ne luetaan karttaan.

```

1 #!/bin/sh
2 #Loputon while-loop, jonka suorittaminen katkaistaan manuaalisesti painamalla CTRL+C
3 while :
4 do
5     #Haetaan uusin TinyWebDB sivu, ja tallennetaan se haluttuun kohteeseen
6     wget -O /home/user/work_dir/current.txt http://1.gpspaikantaja.appspot.com
7     #Käsitellään haettu data, siten, että siivotaan kaikki muu pois paitsi koordinaatit, ja lähetetään koordinaatit points.txt:hen
8     cat /home/user/work_dir/current.txt | cut -d'"' -f2 | grep ^GPS | cut -d" " -f2,3,4 > /home/user/public_html/GPS/source/points.txt
9     echo "Press CTRL+C to stop"
10    sleep 10
11 done
12

```

Kuva 11. Unix-skripti.

Web-sivulla olevan Bing Maps -kartta ja siihen liittyvät elementit on luotu näkyväksi käyttäen pääasiallisesti JavaScript-ohjelmointikieltä. JavaScript ei ilman lisäosia pysty lukemaan palvelimella olevia tiedostoja, joten toteutin koordinaatit sisältävän tiedoston lukemisen käyttämällä PHP-ohjelmointikieltä.

```

1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html>
3 <head>
4 <title></title>
5 <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
6 <script type="text/javascript" src="http://ecn.dev.virtualearth.net/mapcontrol/mapcontrol.ashx?v=7.0"></script>
7 <script type="text/javascript">
8     var map = null;
9     function GetMap()
10     {
11         // Alustetaan kartta
12         map = new Microsoft.Maps.Map(document.getElementById("mapDiv"), {credentials: "AkX4ERU7cEmqS_kamqTsV9vrEAtBA_m_24",
13             center: new Microsoft.Maps.Location(61.498056, 23.760833), //TRE keskusta
14             mapTypeId: Microsoft.Maps.MapTypeId.road,
15             zoom: 13});

```

Kuva 12. Web-sovelluksen alkuosa.

Kuvasta 12 näkyy web-sovelluksen koodin alkuosa. Sivuston runko on toteutettu käyttäen HTML-merkintäkieltä, ja uusi Bing Maps -kartta alustetaan JavaScriptiä käyttäen. Bing Maps Account Centeristä hankittu

ohjelmointirajapinnan käytön mahdollistava sovellusavain on syötetty kohtaan "credentials". Kartan keskipisteeksi on määritetty Tampereen Keskustorin maantieteelliset koordinaatit.

```

17 <?php
18 $file = '/home/user/public_html/GPS/source/points.txt' or die('Could not open file!'); //tiedostopolku määritetty
19 $data = file($file) or die('Could not read file!'); //luetaan tiedosto
20 $coordinates = array();
21
22 foreach ($data as $line) { //lisätään arrayhin jokainen tiedoston rivi omaksi alkioksi
23     $coordinates[] = $line;
24     //echo $line . "<br />";
25 }
26
27 //Luodaan arrayn pohjalta karttapisteet
28 echo "var arrayLocations = new Array();\n";
29 $ix = 0;
30 foreach($coordinates as $key => $value)
31 {
32     echo "arrayLocations[$key] = new Microsoft.Maps.Location($value);\n";
33 }
34 ?>

```

Kuva 13. Web-sovelluksen PHP-osuus.

Kuvassa 13 on web-sovelluksen PHP-osuus. PHP:n avulla TinyWebDB:n tietojen kohdalta koostetun tekstitiedoston jokainen rivi iteroidaan for-lausetta käyttäen omaksi alkiokeeseen JavaScript-taulukkoon. Jokainen taulukon alkio kuvaa yhtä sijaintipistettä kartalla, ja taulukon alkioiden perusteella Bing Maps -karttaan piirretään GPS-paikantimen kulkua kuvaava viiva, jonka luomiseen käytetty JavaScript-koodi on kuvassa 14. Lopuksi kartta tuodaan web-sivulle katseltavaksi.

```

36 // Luodaan polyline
37 var line = new Microsoft.Maps.Polyline(arrayLocations);
38
39 // Piirretään polyline karttaan
40 map.entities.push(line);
41 }
42 </script>
43 </head>
44 <body onload="GetMap();" >
45     <div id='mapDiv' style="position:relative; width:900px; height:900px;"></div>
46 </body>
47 </html>

```

Kuva 14. Web-sovelluksen loppuosa.

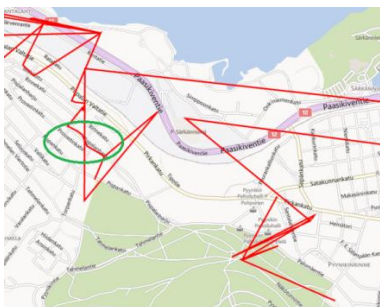
7 OHJELMISTON TESTAUS

7.1 Testaussuunnitelma

Ennen testaamisen aloittamista laadin lyhyen testaussuunnitelman. Testauksessa oli määrä tarkastella ohjelmiston toimivuutta ja toimintavarmuutta erilaisissa olosuhteissa. Testisuunnitelman mukaisesti testejä tehtiin niin taajama-alueilla kuin haja-asutusalueillakin. Päädyin testaamaan sovelluksen toimivuutta aluksi satunnaisesti sovelluksen kehityksen aikana ja lopuksi toteutin testejä ohjelmiston valmiilla versiolla.

7.2 Kehitysversioiden testaaminen

Ennen testaussuunnitelman toteuttamista testasin sovelluksen kehitysversioita. Näiden testausten pohjalta tein jatkuvasti parannuksia sovellukseen. Testasin sovelluksen yleistä toimivuutta kävelen kevättalvella 2013. Sovellus saavutti tasoltaan kohtalaisen jatkuvan paikannustarkkuuden ainoastaan muutamassa testin vaiheessa, joista kuvassa 15 on ympyröitynä vihreällä yksi näistä vaiheista. Muutoin testin aikana sovelluksen paikannus toimi todella epäluotettavasti, ja sovellus paikansi itseään täysin väärin paikkoihin. Tutkittuani asiaa tulin siihen tulokseen, että sovellus ei kykene paikantamaan itseään älykkäästi, kun puhelimessa on päällä asetus, joka sallii puhelimen paikantaa itsensä käyttäen apunaan matkapuhelinverkon ja langattomien verkkojen tukiasemia.



Kuva 15. Sovelluksen kehitysversion testaaminen jalan.

7.3 Testaus kävellessä

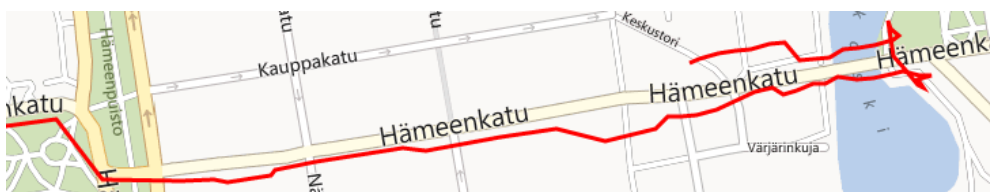
Sovelluksen kehitysversion testauksen jälkeen testasin sovelluksen lopullista versiota uudelleen kävellessä. Tällä kertaa olin kytkenyt puhelimen asetuksista matkapuhelinverkkopaikannuksen pois, jolloin puhelin paikansi itseään enää A-GPS-paikannuksen avulla.

Testin tulokset olivat muutoksen jälkeen huomattavasti parempia, kuten kuva 16 osoittaa. Testin aikana matkapuhelin oli sijoitettuna taskuun, joten sillä ei ollut testin aikana suoraa näkyvyyttä taivaalle. Tästä huolimatta se kykeni tarkkaan paikannukseen. Testituloksesta käy kuitenkin ilmi, että paikannuksessa on noin metrin epätarkkuus.



Kuva 16. Ensimmäinen testaus kävellessä.

Järjestelmän toimivuutta testattiin kävellessä myös kaupunkiolosuhteissa, jonka tuloksena on kuva 17. Matkapuhelin oli sijoitettu jälleen taskuun, ja matkapuhelin paikansi itseään käyttäen A-GPS-paikannusta. Testituloksesta on havaittavissa, että paikannussovellus kykeni keskustaolosuhteissa, korkeiden rakennusten ympäröimänä paikantamaan vain kohtalaisesti. Paikannustarkkuus ei ole aivan yhtä hyvä kuin keskusta-alueen ulkopuolella, mutta paikantimen todellinen reitti on silti suhteellisen hyvin nähtävissä.

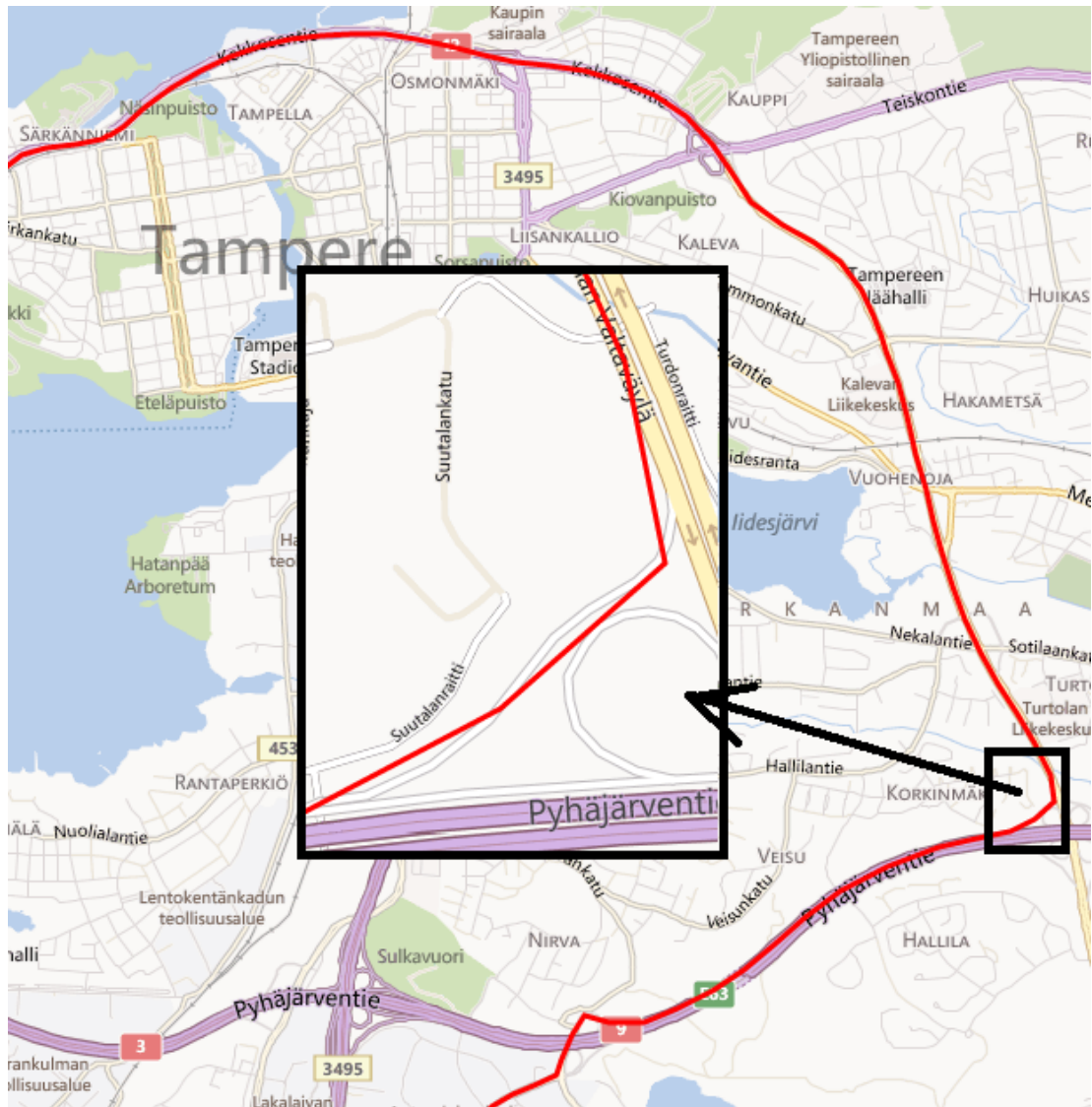


Kuva 17. Toinen testaus kävellessä.

7.4 Testaus ajoneuvoissa

Testasin sovelluksen toimintaa useissa eri ajoneuvoissa. Halusin nähdä mikä merkitys sovelluksen paikannustarkkuuteen ajoneuvon nopeudella ja matkapuhelinsignaalin vahvuudella on sovelluksen toimintaan.

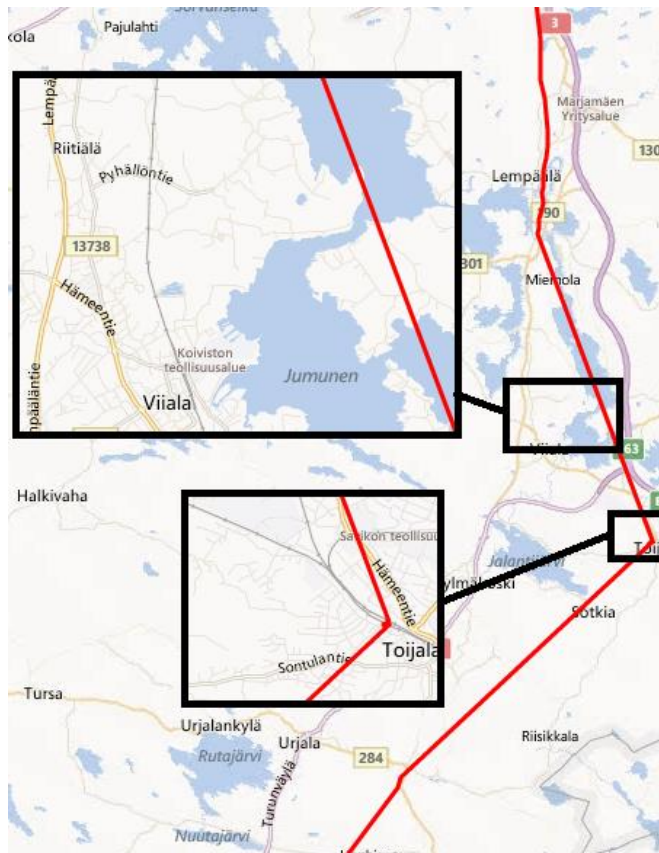
Ensimmäinen ajoneuvo, jossa testasin sovellusta oli henkilöauto. GPS-paikannussovelluksen ollessa päällä älypuhelin oli sijoitettuna henkilöauton kojelaudan päälle siten, että se saavutti hyvän näkyvyyden taivaalle. Testi suoritettiin hyvissä sääolosuhteissa, ja testin aikana ajoneuvon nopeus vaihteli 0 – 100 km/h välillä. Testireitti kulki pääasiassa aukeilla valta- ja moottoritieosuuksilla, joten ympäristön rakenteet eivät häirinneet GPS:n toimintaa. Lisäksi reitin varrella oli pääasiassa erinomainen matkapuhelinverkon kuuluvuus, mikä edesauttoi A-GPS:n toimintaa.



Kuva 18. Sovelluksen testaus henkilöautossa.

Henkilöautossa suoritetun testin tulos on nähtävissä kuvasta 18, josta käy ilmi, että sovelluksen toimintavarmuus hyvällä taivasnäkyvyydellä on taajamaolosuhteissa erinomainen. Sovellus kykeni taajamaolosuhteissa poikkeuksetta vastaanottamaan GPS-satelliitilta uuden signaalin ja lähettämään oman sijaintinsa palvelimelle ohjelman minimiväliajaksi määritetyn 10 sekunnin välein.

Testin toinen osuus suoritettiin VR:n InterCity2-matkustajajunassa matkalla Tampereelta Turkuun, jolloin matkapuhelin oli sijoitettuna ikkunan viereen siten, että sillä oli kohtalainen yhteys taivaalle. Sääolosuhteet olivat melko huonot; taivaalla oli runsaasti pilviä ja vesisadetta. Lisäksi juna saavutti matkalla ajoittain suuria nopeuksia. Juna kulki suurilta osin taajama-alueiden ulkopuolella, alueilla, joilla matkapuhelinverkon kuuluvuus on erityisen heikko, mikä vaikutti todennäköisesti heikentävästi matkapuhelimen A-GPS-paikannussensorin toimintaan.



Kuva 19. Sovelluksen testaus junassa.

Kuvassa 19 on ote junassa suoritetusta testistä. Sovellus pystyi paikantamaan reittinsä suhteellisen hyvin Tampereen rautatieasemalta Lempäälään asti, mutta sen jälkeen sen toimintavarmuus laski. Sovellus piti useiden minuuttien taukoja, kun juna kulki alueilla, joilla matkapuhelinverkon kuuluvuus on heikko. Sovellus sai paikkatiedon ainoastaan saavuttaessa taajama-alueilla sijaitseville rautatieasemille, kuten kuvassa Toijalan asemalla.

8 POHDINTA JA KEHITYSIDEAT

Opinnäytetyön päätavoitteena oli GPS-paikannusohjelmiston suunnittelu, toteutus ja testaus. Ohjelmiston toteutus ja testaus onnistuivat suunnitelluilla tavoilla.

Testauksen aikana havaitsin lukuisia parannusehdotuksia ja kehitysideoita. Kehitysideat liittyvät pääasiassa ohjelmiston toimintavarmuuden ja käyttäjäkokemuksen parantamiseen.

Testit onnistuivat hyvin ja testitulokset olivat pääasiassa vakuuttavia. Suurin testeissä havaittu ongelma koski ohjelman toimintavarmuutta, sillä paikannussovellus ei toiminut halutulla tavalla heikkojen matkapuhelinverkko-yhteyksien takia. Ohjelmiston toimintavarmuutta korkeiden rakennusten läheisyydessä ja syrjäseuduilla voisi parantaa ajoneuvoon kiinteästi asennetun antennin avulla, joka parantaisi GPS-signaalia ja matkapuhelinverkon signaalia.

Ohjelmiston jatkokehityksen kannalta tärkeintä olisi luoda ohjelmistoon mahdollisuus kirjautua paikannussovellukseen ja seurantasovellukseen oman käyttäjätunnuksen avulla. Tämä mahdollistaisi ohjelmiston käytön useammalle, kuin yhdelle käyttäjälle. Opinnäytetyössä kehitettyä versiota ohjelmasta ei ole vielä mahdollista käyttää kuin yhden käyttäjän kerrallaan. Mikäli paikannussovelluksen paikannustoiminto on samanaikaisesti päällä kahdessa eri matkapuhelimessa, on seurauksena päällekkäisten merkintöjen syntyminen TinyWebDB-tietokantaan, mistä johtuen seurantasovelluksesta pystyy seuraamaan vain toista GPS-paikantajaa.

Ohjelmistoon voi myös lisätä ominaisuuksia, kuten paikantimen nopeusseurannan, sekä tiedot paikantimen korkeudesta. Testivaiheen paikannussovellus ilmoitti ainoastaan viimeisimmän sijaintinsa ilman nopeus- tai korkeustietoja.

Uutena ominaisuutena paikannussovellukseen voisi myös olla toiminto, jolla pystyy vaikuttamaan paikkatietojen kirjaamistiheyteen. Nykyinen 10 sekuntia on riittävä useimmissa tilanteissa, mutta esimerkiksi kaupungissa liikkuesssa, kun käännöksiä saattaa tulla nopeasti useita 10 sekunnin kirjaamistiheys ei ole riittävä.

Jatkokehitystä ohjelmistosta voi räätälöidä esimerkiksi kuljetusyritykselle ajoneuvoseurantajärjestelmän tai etäisyysmittarin osaksi golfin harrastajille suunniteltua matkapuhelinsovellusta.

Opinnäytetyön paikannussovelluksen kehitykseen käytetty App Inventor for Android on hyvä työkalu, jos haluaa päästä tutkimaan, mitä kaikkea omalla Android-älypuhelimella voi tehdä. Se mahdollistaa Android-älypuhelimien ominaisuuksien hyödyntämisen tavalla, joka ei ehkä muuten olisi käyttäjän haluamalla tavalla mahdollista.

LÄHTEET

- Airos, E., Korhonen, R., Pulkkinen, T. 2007, Satelliittipaikannusjärjestelmät, Helsinki: Edita Prima Oy. Viitattu 12.3.2013 <http://www.puolustusvoimat.fi/wcm/658d1080428c8d79900bd2e60feb2862/PVTTJulkaisu12.pdf?MOD=AJPERES>.
- Android (operating system) 2013. Wikipedia. Viitattu 19.3.2013. [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)).
- Android.fi 2010. Mikä ihmeen Android? Viitattu 15.3.2013. <http://www.android.fi/mika-ihmeen-android/>.
- Bing Maps Account Center. Create a Bing Maps Account. Viitattu 30.4.2013. <http://bingmapsportal.com/>.
- Cooking Hacks 2013a. Arduino Uno Rev.3. Viitattu 19.3.2013. <http://www.cooking-hacks.com/index.php/arduino-uno.html>.
- Cooking Hacks 2013b. 3G/GPRS Shield For Arduino / Raspberry Pi (3G + GPS). Viitattu 19.3.2013. <http://www.cooking-hacks.com/index.php/3g-gprs-shield-for-arduino-3g-gps.html>.
- Cooking Hacks 2014c. 7.4V LiPo 2200mAh Battery (Arduino Power Jack). Viitattu 19.3.2013. <http://www.cooking-hacks.com/index.php/7-4v-lipo-2200mah-battery-arduino-power-jack.html>.
- Cooking Hacks 2014d. Arduino Project Enclosure. Viitattu 19.3.2013. <http://www.cooking-hacks.com/index.php/arduino-project-enclosure.html>.
- Eadicicco, L. 2012. Digital Trends 4.12.2012. Viitattu 15.3.2013. <http://www.digitaltrends.com/mobile/android-most-popular-platform-2016/>.
- Google 2013. What is (isn't) Google App Engine 19.3.2013. Viitattu 23.4.2013 <https://developers.google.com/appengine/training/intro/whatisgae>.
- GPS.gov 2012. Data From the First Week Without Selective Availability. Viitattu 2.2.2013. <http://www.gps.gov/systems/gps/modernization/sa/data/>.
- GPS Daily 2010. China's Compass Finds Route To Rival GPS 19.1.2010. Viitattu 22.4.2013 http://www.gpsdaily.com/reports/China_Compass_Finds_Route_To_Rival_GPS_999.html.
- GSM Arena 2013. Samsung I9105 Galaxy S II Plus. Viitattu 24.4.2013 http://www.gsmarena.com/samsung_i9105_galaxy_s_ii_plus-5213.php.
- James, R. 2009. GPS. Time 26.5.2009. Viitattu 4.2.2013. <http://www.time.com/time/magazine/article/0,9171,1901500,00.html>.
- Junat kartalla - VR 2011. Junat kartalla. Viitattu 26.4.2013. <http://www.vr.fi/fi/index/aikataulut/junatkartalla.html>.
- Ladyada.net 2011. GPS Shield. Viitattu 17.3.2013. <http://www.ladyada.net/make/gpsshield/index.html>.
- Luontoon.fi 2011. Geokätköily. Viitattu 7.5.2013. <http://www.luontoon.fi/harrastukset/geokatkoily/Sivut/Default.aspx>.

- MIT App Inventor 2013a. Creating a Custom TinyWebDB Service. Viitattu 15.4.2013. <http://beta.appinventor.mit.edu/learn/reference/other/tinywebdb.html>.
- My-car-computer.com 2012. Trilateration. Viitattu 9.3.2013 <http://www.my-car-computer.com/trilateration.html>.
- Navicom 2013. Ajoneuvon seurantatiedot. Viitattu 29.4.2013 <http://www.navicom.fi/seuranta/ajoneuvoseuranta>.
- O'Brien, K. 2010. Smartphone Sales Taking Toll on G.P.S. Devices. The New York Times 14.11.2010. Viitattu 2.5.2013. http://www.nytimes.com/2010/11/15/technology/15iht-navigate.html?_r=1&.
- Ogaja, C. 2011. Applied GPS for Engineers and Project Managers. Virginia: ASCE.
- Reuters 2012. GPS attacks risk maritime disaster, trading chaos 21.2.2012. Viitattu 23.4.2013 <http://www.reuters.com/article/2012/02/22/us-security-gps-idUSTRE81L00E20120222>.
- Rouse, M. 2011. WYSIWYG (what you see is what you get). WhatIs.com. Viitattu 9.5.2013. <http://whatIs.techtarget.com/definition/WYSIWYG-what-you-see-is-what-you-get>.
- SearchCloudProvider 2011. Cloud services. Viitattu 9.5.2013. <http://searchcloudprovider.techtarget.com/definition/cloud-services>.
- SearchEnterpriseLinux 2006. PHP. Viitattu 9.5.2013. <http://searchenterpriselinix.techtarget.com/definition/PHP>.
- SearchExchange 2010. Application program interface (API). Viitattu 9.5.2013. <http://searchexchange.techtarget.com/definition/application-program-interface>.
- SearchSOA 2005a. HTML (Hypertext Markup Language). Viitattu 9.5.2013. <http://searchsoa.techtarget.com/definition/HTML>.
- SearchSOA 2005b. JavaScript. Viitattu 9.5.2013. <http://searchsoa.techtarget.com/definition/JavaScript>.
- Sulopuisto, O. 2009. Kulutus laskuun GPS:n ja liikenteen ennustamisen avulla. Digitoday 19.2.2009. Viitattu 16.4.2013 <http://www.digitoday.fi/tiede-ja-teknologia/2009/02/19/kulutus-laskuun-gpsn-ja-liikenteen-ennustamisen-avulla/20094584/66>.
- Tampereen joukkoliikenne 2013. Lissu liikenteenseuranta. Viitattu 2.5.2013. <http://lissu.tampere.fi/>
- TechTerms.com 2008. Emulation. Viitattu 18.5.2013. <http://www.techterms.com/definition/emulation>.
- U.S. Naval Observatory 2013. Current GPS constellation. Viitattu 4.2.2013. <http://tycho.usno.navy.mil/gpscurr.html>.
- Verkkokauppa.com 2013. LG Optimus L5 Android-puhelin, valkoinen. Viitattu 12.4.2013 <http://www.verkkokauppa.com/fi/product/8944/dgsdk/LG-Optimus-L5-Android-puhelin-valkoinen>.
- Windows Phone Central 2009. GPS vs. aGPS: A Quick Tutorial. Viitattu 3.5.2013. <http://www.wpcentral.com/gps-vs-agps-quick-tutorial>.